Dell™ OpenManage™ Client Instrumentation Version 7.6 User's Guide

Notes and Notices

NOTE: A NOTE indicates important information that helps you make better use of your computer.

NOTICE: A NOTICE indicates either potential damage to hardware or loss of data and tells you how to avoid the problem.

Contents

1

Introduction

The Dell™ OpenManage™ Client Instrumentation (OMCI) software enables remote management application programs to access client system information, monitor the client system status, or change the state of the client system; such as shutting it down remotely. Through standard interfaces, OMCI exposes key system parameters allowing administrators to manage inventory, monitor system health, and gather information on deployed Dell client systems.

OMCI enables Dell client systems to be managed using the Common Information Model (CIM) standard. CIM reduces your total cost of ownership, improves security, and provides a holistic approach to managing all devices in the enterprise, including clients, servers, storage, network, and software devices.

CIM Overview

The CIM, created by the Desktop Management Task Force (DMTF) as part of the Web-based Enterprise Management (WBEM) initiative, provides a unified view of physical and logical objects in the managed environment. CIM is a systems management protocol defined by industry standards.

The following are important CIM details:

- CIM is an object-oriented data model for describing management information. CIM describes the way the data is organized, not necessarily the transport model used to transport the data. The most prevalent transport method is the Microsoft® Windows® Management Instrumentation (WMI).

- CIM-capable management applications gather information from a variety of CIM objects and devices, including client and server systems, network infrastructure devices, and applications.

- The CIM specification details mapping techniques for improved compatibility with other management protocols.

- The CIM data model abstracts and describes all elements in a network environment. The CIM schema provides the actual data model descriptions and arranges the network into a series of managed objects, all interrelated and broadly classified.
- The CIM schema is defined by the Managed Object Format (MOF) file, which provides a standardized model for describing management information between clients in a management system. The MOF file is not bound to a particular implementation, and it allows the interchange of management information between many different management systems and clients.

WMI Overview

WMI is Microsoft's implementation of CIM instrumentation. WMI supports CIM and Microsoft-specific CIM extensions. The WMI management infrastructure also includes the CIM Object Manager (CIMOM), which is the interface and manipulation point for CIM objects and information. It acts as a facilitator in gathering information and manipulating object properties. OMCI provides data to WMI, which is the common interface to WMI management applications. For more information, see "Windows Management Instrumentation."

Data Sources Used by OMCI

OMCI contains the underlying driver set that collects system information from a number of different sources on the client system, including the BIOS, CMOS, System Management BIOS (SMBIOS), System Management Interface (SMI), operating system, Application Programming Interfaces (APIs), Dynamic Link Libraries (DLLs), and registry settings. OMCI exposes that information through the CIMOM interface of the WMI stack and enables IT administrators to remotely collect asset information, modify CMOS settings, receive proactive notification of potential fault conditions, and be alerted to potential security breaches.

OMCI uses a variety of data sources resident on the client system which includes:

*   Microsoft Win32 WMI providers
*   SMBIOS
*   Device drivers
*   Operating system registry
*   Operating system Application Programming Interfaces (APIs)
*   WMI repository

OMCI Architecture

The OMCI architecture is based on a layered model that is tightly integrated with the Microsoft WMI stack as shown in Figure 1-1.

*   **WMI application layer** — Consists of management applications such as Dell Client Manager, Dell OpenManage Client Administrator (OMCA), and other standards-based management tools and WMI applications such as Microsoft SMS, LANDesk, and so on. This layer is not a part of OMCI and the applications are consumers of systems management data supplied from the client by OMCI. These applications request client information and receive client alerts through the CIMOM.

*   **Dell WMI provider** — Lies beneath the CIMOM and contains two CIM providers which register with the CIMOM. When the CIMOM receives a request for information, it routes the request to the appropriate provider. Both Dell and Microsoft providers exist in this layer, and they provide information on system devices. The providers send management application requests from the CIMOM to the data router.

*   **Data router** — Collects information from system components in the data provider layer and passes the information up to the Dell WMI provider layer, which then presents it to the CIMOM.

*   **Data provider** — Provides data to the data router. When it receives a request from the data router, the data provider queries the client system's hardware, drivers, and operating system to determine the available instrumentation. It then applies all platform and customer overrides to the data before returning it to the data router.

For example, a management console in the WMI application layer requests the available free space on the client system hard drive. The WMI application layer makes the request over the network to the CIMOM (in the Dell WMI provider layer) on the client system. The CIMOM routes the request to the appropriate WMI provider, which, in turn, routes the request through the data router layer to the disk data provider in the data provider layer. The information is then returned (through the same path in reverse) to the management console.

> **NOTE:** OMCI 7.6 and later versions will not be supported by Dell OpenManage IT Assistant (ITA) 8.*x*.

Figure 1-1.   Dell OpenManage Client Instrumentation Architecture
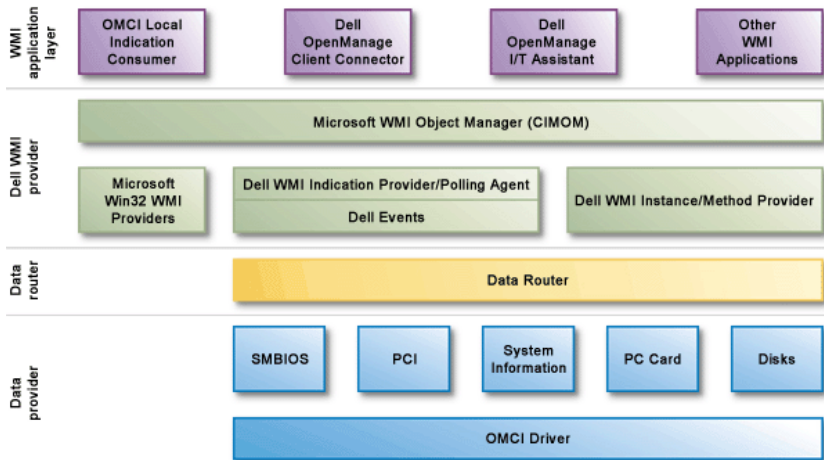
Features

The following are key features of OMCI:

- CIM 2.6 Schema Support
- Dell LegacySelect 2.0 Support
- Remote Boot Device Select
- WMI Security
- Event Reporting
- Remote Shutdown
- Remote Wake-Up Configurability
- WfM Version 2.0 Compliance

CIM 2.6 Schema Support

CIM is a standard that describes management data that can be used by many management application programs. OMCI conforms to the CIM 2.6 Schema, and includes two WMI providers: Dell WMI Indication Provider/Polling Agent and Dell WMI Instance/Method Provider.

Dell LegacySelect 2.0 Support

LegacySelect allows a network administrator to selectively enable or disable legacy components in a Dell system. It is a BIOS-level interface which lets you control the level of legacy technology your varied workgroups require. LegacySelect 2.0 supports the ability to enable or disable ports, including USB, PCI slots, and disk drives.

The LegacySelect 2.0 specification supports the following components:

- Serial port
- Parallel port
- Diskette drive controller
- Personal System/2 (PS/2) mouse port
- Integrated audio controller
- Integrated network interface controller (NIC)
- Universal Serial Bus (USB) controller
- Peripheral Component Interconnect (PCI) slot

A network administrator can change the settings of these components either through the computer's system setup or by using a WMI management application.

NOTICE: Changing **LegacySelect** attributes such as **Integrated Drive Electronics (IDE) Controller** may cause a computer system to become unusable. If this occurs, enter system setup on the client system and change the applicable setting.

Remote Boot Device Select

Remote Boot Device Select allows a network administrator to remotely configure the order of devices from which the system attempts to boot. It consists of two groups: **Hard Disk Drive Sequence** and **Boot Device Sequence**. **Hard Disk Drive Sequence** is used to set the order of the hard drives, and **Boot Device Sequence** is used to set the order of the type of boot device, such as hard drive, NIC, CD-ROM drive, and diskette drive.

WMI Security

WMI provides for user authentication before granting access to CIM data and methods. Access privileges are enforced by distributed component object model (DCOM) security and the CIMOM.

Event Reporting

OMCI detects events on Dell systems and can alert the local user and network administrator to potential failures, configuration changes, and chassis intrusions. These events are displayed by a system management application program such as IT Assistant.

OMCI allows you to set polling intervals for a client Dell computer's Self-Monitoring Analysis and Reporting Technology (SMART) hard drive, environmental monitors, chassis intrusion, hard drive threshold, or error checking and correction (ECC) code single-bit memory error events.

A polling interval defines how often the instrumentation code checks the device status. Polling intervals are important because they determine how often OMCI checks for an alert state on the system, and therefore, how often it generates an alert. If the polling intervals are set too high, you may miss an alert. For example, systems that support a thermal probe also have a maximum temperature where the system will shut down regardless of other

processes that may be running. If the temperature increases too quickly and OMCI does not poll the system in time, it will not generate a shutdown message and the system will not shut down cleanly.

OMCI also contains a start delay for the polling intervals. This determines how long after the operating system loads, OMCI starts checking for the alert status. An alert may be missed if the start delay is set too high.

For instructions on setting polling intervals and enabling or disabling alerts for these devices, see the documentation for your systems management application program.

Remote Shutdown

OMCI supports remote system shutdown and restart.

Remote Wake-Up Configurability

OMCI supports configuration of remote wake-up settings. Remote wake-up is a function of the client system and NIC.

WfM Version 2.0 Compliance

OMCI is compliant with the Intel® Wired for Management (WfM) 2.0 standard. For more information on this standard and its specifications, see the Intel website at **www.intel.com.**

What's New

The following are new updates to OMCI 7.6:

- **Support for new platforms:** Support for new Dell Latitude™, Dell OptiPlex™, and Dell Precision™ systems to be released in FY09.

- **Disable boot from an USB device:** Use the USB "No Boot" option available in the USBPorts property of the Dell_SMBIOSSettings class to turn off the ability to boot from an USB device.

- **Additional PCI Device Information:** Earlier releases of OMCI enabled you to get limited information about PCI devices such as only the IDs of systems, subsystems, vendors, and subvendors. OMCI 7.6 provides more information such as the PCI device manufacturer name and the model name in the client system.

- **Additional Display Device Information:** Earlier OMCI releases provided limited display device information. OMCI 7.6 also provides the manufacture date in <yyyy Week ww> format; y-year, w-week number (for example, 2008 Week 16) and the possible sleep states of the monitor.

- **Additional Physical Memory Information:** Previous OMCI releases retrieved the value of the memory capacity property in the Dell_PhysicalMemory class. OMCI 7.6 also retrieves the Manufacturer, Memory Type, and Speed property values of the Dell_ PhysicalMemory class.

- **Additional Logical Disk Information:** Previous OMCI releases contained logical disk information in the Dell_Logicaldisk class such as size, description, free space, path, and so on. OMCI 7.6 also retrieves the Compressed (indicates whether or not the logical volume exists as a single compressed entity) and FileSystem(indicates the file system on the logical disk) property values of Dell_Logicaldisk class in the WMI inventory of the client system.

NOTE: NTFS file system compression is not supported. If NTFS file based compression is supported, the **Compressed** property is FALSE.

- **Added support for BIOS settings:** BIOS settings listed in "BIOS Settings Supported in OMCI 7.6" and marked with an asterisk (*) are added in OMCI 7.6.

Supported Operating Systems

The following operating systems are supported for installing OMCI:

- Microsoft Windows 2000 Professional SP4
- Microsoft Windows XP Professional SP2 or later x86 (32-bit) and x64 (64-bit) versions
- Microsoft Windows Vista® Business x86 (32-bit) and x64 (64-bit) versions
- Microsoft Windows Vista Ultimate x86 (32-bit) and x64 (64-bit) versions
- Microsoft Windows Vista Enterprise x86 (32-bit) and x64 (64-bit) versions

Latest Updates

For the latest update information about OMCI, see the **readme** file located in the installation directory. The default installation path for a typical installation of OMCI is **C:\Program Files\Dell\OpenManage\Client**.

Other Dell Documents You Might Need

In addition to this *User's Guide*, you can find the following guides either on the Dell Support website or on the *Systems Management Tools and Documentation* DVD:

- The *Dell OpenManage Client Instrumentation Reference Guide* provides detailed information on all OMCI classes, properties, and their descriptions.

- The *Dell OpenManage Client Connector User's Guide* provides information on how to view and configure OMCI properties and perform remote management functions.

- The *Dell Client Configuration Utility User's Guide* provides information on how to obtain or configure BIOS settings. You can restart or shut down the system or flash the BIOS with Dell Client Configuration Utility. The utility creates an executable that can be run as a standalone package or used with software deployment tools to execute remotely across an enterprise.

Other Resources Which Might Be Of Interest

- The Dell OpenManage Client Administrator (OMCA), an integrated suite of client management applications developed in partnership with Altiris®, to provide optimized client system management, can be downloaded from the Altiris website.

- Dell Client Manager (DCM) which includes Dell-specific hardware and BIOS inventory information, helps configure and update BIOS, supports remote shutdown and remote wakeup, and provides hardware health monitoring.

Obtaining Technical Assistance

If at any time you do not understand a procedure described in this guide, or if your product does not perform as expected, different types of help are available. For more information see "Getting Help" in your system's *Setup Guide*.

Additionally, Dell Enterprise Training and Certification is available; see **www.dell.com/training** for more information. This service may not be offered in all locations.

2

Installing Dell™ OpenManage™ Client Instrumentation

Dell OpenManage Client Instrumentation (OMCI) can be installed locally using the OMCI installation wizard, or remotely through a silent or administrative installation.

Local Installation: The OMCI installation wizard allows you to install the complete package or to do a custom installation of specific components required for a particular environment. OMCI can also be modified, repaired, or removed through the **Add/Remove Programs** dialog in the Microsoft® Windows® operating system environment.

Silent Installation: OMCI provides a series of command-line switches for silent installation. This allows you to remotely install the product using an electronic software distribution tool with no end-user interaction.

Administrative Installation: You can use a server image to deploy OMCI to client computers on a network by using a login script, Windows system policies, or other methods.

A default OMCI uninstallation does not uninstall the driver files and the associated **DLL** files and they remain on the system. A command line property (UNINSTALL_DRIVER=1) has to be specified on the command line for a command line product uninstall, which will uninstall OMCI as well as the driver files.

For example, the following command uninstalls the product as well as the driver files:

```
msiexec.exe /X{09ADA709-10D2-4C38-807C-7B84F696F6CE}
/qn UNINSTALL_DRIVER=1
```

📝 **NOTE:** To install OMCI, you must have administrative rights on the client system. This means that you must be authenticated on the client system as a user who is a member of the Administrators group, typically the Administrator. The target system must be a Dell system with SMBIOS version 2.3 or later. Otherwise, the OMCI 7.6 installer will exit without installing OMCI.

NOTE: You cannot upgrade from OMCI 6.*x* to OMCI 7.6 using the OMCI 7.6 installer if the target system has OMCI 6.*x* on it. Ensure that you uninstall OMCI 6.*x* before installing OMCI 7.6. OMCI version 7.2 and later can be installed only on Dell systems running the Windows operating systems listed in the "Supported Operating Systems" section.

NOTE: You can upgrade to OMCI 7.6 from OMCI versions 7.4 A00, 7.5 A00, 7.5 A01, 7.5 A02, and 7.5 A03. If the upgrade path includes an operating system upgrade to Windows Vista, then you must upgrade OMCI to 7.6 first, and then upgrade the operating system to Windows Vista; vice versa is not supported.

NOTE: Separate installation programs exist for Windows 32- and 64-bit Edition operating systems, and are available for download from the Dell Support website at **support.dell.com**.

User Account Control in Windows Vista

In previous versions of Windows, user accounts were often members of the local Administrators group and had administrator privileges. Members of the local Administrators group could install, update, and run software since an administrator account had system-wide access. When a user was added to the local Administrators group, that user was automatically granted every Windows privilege because administrator privileges provide access to all operating system resources. User accounts with administrator privileges posed a security risk by providing access to operating system resources that can be exploited by malicious software (or malware).

In Windows Vista, there are two types of user accounts: standard user accounts and administrator accounts. Additionally, Windows Vista provides User Account Control (UAC) to improve system security by limiting the operating system access available to standard user accounts and requiring members of the Administrators group (except the Administrator account) to acknowledge the elevation of privileges when performing certain actions. The primary difference between a standard user and an administrator in Windows Vista is the level of access the user has over core, protected areas of the computer. With UAC enabled, Windows Vista either prompts for consent to elevate privileges (in the case of a user in the Administrators group) or for administrator credentials to elevate privileges (in the case of all other users) before launching a program or task that requires full administrator access. UAC prompts members in the Administrators group (except the Administrator account) to click **Continue**, if they need to elevate privileges,

or to click **Cancel** when performing functions that may entail a security risk. With UAC, you can run most applications, components and processes with a limited privilege, but have elevation potential for specific administrative tasks and application functions.

During OMCI installation or uninstallation, depending upon the access level of the user, Windows Vista prompts the user for administrator authorization. You must provide the administrator password for installing or uninstalling OMCI if you have standard account privileges, since OMCI installation and uninstallation need administrative privileges. Even when you are logged in as a member of the Administrators group (except the Administrator account), UAC requires you to give your consent to allow the installation or uninstallation.

NOTE: Since the user experience is configurable with the Security Policy Manager snap-in (**secpol.msc**) and with Group Policy, there are multiple UAC user experiences. The configuration choices made in your environment will affect the prompts and dialogs seen by standard users, administrators, or both. The specific behavior of the UAC elevation prompt is dependent upon the Security Policy setting.

Local Installation

Installing the Client

1   Click **Download Now** to download the installation file (for example, r118302_a00.exe) from the Dell Support website at **support.dell.com**. When the **File Download** window appears, save the file.

   The installation file contains the components for the following languages: English, French, German, Spanish, Simplified Chinese, and Japanese. It can be installed only on systems running the Windows operating systems listed in the "Supported Operating Systems" section.

2   Double-click the downloaded installation file. The **Systems Management** dialog box is displayed with information on the software version, supported systems, supported operating systems, and supported languages.

3   In the **Systems Management** dialog box, click **Continue**. Specify the folder where you want to unzip the installation file and click **OK**. A confirmation message appears after the file is successfully unzipped. Click **OK**.

4   Double-click the extracted installation executable file, **setup.exe**. The **Choose Setup Language** dialog box appears. Select a language and click **OK**.

5 When the **InstallShield Wizard** window displays the `Welcome` message, click **Next**.

The **InstallShield Wizard** window displays the message `Setup Type`.

6 Choose the installation type by clicking one of the following options:

- **Complete** — Installs all modules and WMI support to a default directory.

- **Custom** — Enables you to install specific program features and specify the installation directory.

7 Select the **Complete** option and click **Next**. The **InstallShield Wizard** window displays the message `Ready to Install the Program`. If the option **Custom** is selected, the **Custom Setup** window is displayed, allowing you to install specific program features and also specify the directory in which you want to install OMCI. Click **Next**.

8 Click **Install** to begin the installation.

If OMCI version 6.x is installed on the system, the installation program prompts you to uninstall it and then run the installation program again.

The **InstallShield Wizard** window displays the message `Installing OMCI`. A progress bar indicates the installation status. When the installation is completed, the message changes to `InstallShield Wizard Completed`.

9 Click **Finish** to end the installation.

Uninstalling the Client

To remove the client instrumentation using the **Add/Remove Programs** icon on the **Control Panel**, perform the following steps:

1 Click the **Start** button, select **Settings**, and click **Control Panel**.

2 Select **Add/Remove Programs**.

3 From the **Add/Remove Programs** dialog box, select **OMCI** and click **Remove**.

4 Follow the instructions on the screen.

✎ **NOTE:** In Windows Vista, use the **Programs and Features** icon on the Control Panel for installing or uninstalling OMCI.

Silent Installation

Installing the Client in the Silent Mode

In the silent mode (no dialog boxes appear), you can perform complete and custom installations.

NOTE: When typing commands, enter spaces exactly as shown in the examples in this section.

Performing a Complete Installation in Silent Mode

To perform a silent unattended installation of OMCI with all components in the default directory:

```
setup.exe /s /v/qn
```

where **setup.exe** is the installation executable file for OMCI.

To specify the installation language use the `/L<language ID>` option, where `<language ID>` is **1031** (German), **1033** (English), **1034** (Spanish), **1041** (Japanese), **1036** (French), or **2052** (Simplified Chinese). If the installation language is not specified, the installer selects the default operating system language, or to English if OMCI does not support the operating system language.

For example, the following command line installs the French version in silent mode:

```
setup.exe /s /L1036 /v/qn
```

Performing a Custom Installation in Silent Mode

To specify a custom installation directory, modify the command line as follows:

```
setup.exe /s /v"/qn INSTALLDIR=<destination>"
```

where `<destination>` is the custom directory and `INSTALLDIR` must be in uppercase.

For example, the following command line installs OMCI with all components in silent mode under the directory **c:\destination**:

```
setup.exe /s /v"/qn INSTALLDIR=c:\destination"
```

Upgrading the Client in Silent Mode

To upgrade from one version of OMCI 7.x to a later version of OMCI 7.x, execute the following command:

```
setup.exe /s /v"/qn REINSTALLMODE=voums REINSTALL=
ALL"
```

A reboot might be required for the upgrade. To suppress it, modify the command line as follows:

```
setup.exe /s /v"/qn REINSTALLMODE=voums REINSTALL=ALL
REBOOT=REALLYSUPPRESS"
```

Uninstalling the Client in Silent Mode

You can uninstall the client instrumentation in silent mode either using or without using the installation program.

To uninstall the client instrumentation without the installation program in silent mode, run one of the following commands:

For all 32-bit Windows installations:

```
msiexec.exe /X{73F1BDB7-11E1-11D5-9DC6-00C04F2FC33B}
/qn
```

For all 64-bit Windows installations:

```
msiexec.exe /X{09ADA709-10D2-4C38-807C-7B84F696F6CE}
/qn
```

To uninstall the client instrumentation with the installation program in silent mode, run the following command from the directory containing the installation program:

```
setup.exe /s /x /v/qn
```

Administrative Installation

To create a server image of OMCI on your network using an administrative installation, perform the following steps:

1 Using the OMCI installation package, run the following command:

    setup.exe /a

    The **Choose Setup Language** dialog box appears.

2 Select a language and click **OK**.

3 When the **InstallShield Wizard** window displays the `Welcome` message, click **Next**.

4 Enter the network location in the **Network Location** dialog box and click **Install** to begin the installation.

    The **InstallShield Wizard** window displays the message `Installing OMCI`. A progress bar indicates the installation status. When the installation is completed, the message changes to `InstallShield Wizard Completed`.

5 Click **Finish** to end the installation.

You can use the server image to deploy OMCI to client systems on your network by using a login script, Windows system policies, or other methods.

For example, the following command installs OMCI using the server image in silent mode:

msiexec /i <Network Location>\omci.msi /qn

where `<Network Location>` is the path to the server image.

Ø **NOTE:** For information on using the server image to deploy OMCI, see the documentation for the application program that you are using as your deployment agent.

3

Windows Management Instrumentation

WMI Overview

Microsoft® Windows® Management Instrumentation (WMI) is Microsoft's implementation of the Web-Based Enterprise Management (WBEM) effort. It is implemented on Microsoft Windows platforms.

There are two components to WMI. In the Dell™ OpenManage™ Client Instrumentation (OMCI) architecture (see Figure 1-1), both the components are represented by the Microsoft WMI Object Manager. The first component is the Common Information Model (CIM) Object Manager (CIMOM), which is the interface and manipulation point for CIM objects and information. It acts as a facilitator in gathering information and manipulating object properties. Microsoft has implemented this component as the Windows management (winmgmt) service. The CIMOM is a software middle layer that mediates interactions between high-level management applications and the lower levels of instrumentation, such as OMCI and other providers. The CIMOM ensures that data supplied by providers is presented to management applications in a uniform and provider-independent way. The CIMOM does this, in part, by using the Component Object Model (COM) Application Programming Interface (API) for communication with providers and management applications.

The second WMI component is the repository. The repository is a binary file where the CIMOM stores management data. Such data includes information from the compiled Managed Object Format (MOF) file(s), including the CIM class definitions, properties, qualifiers, and hierarchical relationships. Instance data, as it becomes available, is also stored here.

WMI Security

WMI provides for user authentication before granting access to CIM data and methods. Access privileges are enforced by Distributed Component Object Model (DCOM) security and the CIMOM. Access, whether full

or limited, can be granted to users on per-namespace basis. There is no implementation of class or property-level security. By default, users that are members of the **administrators** group have full local and remote access to WMI.

WMI security can be configured using the WMI Control available in the Computer Management console under the **Services and Applications** section. Right-click **WMI Control**, and then click **Properties**. You can configure namespace-specific security from the **Security** tab. WMI Control can also be executed from the **Start** menu or the command prompt by running **wmimgmt.msc**.

NOTE: To connect remotely to WMI services, you must have administrative rights for both the local and the remote systems.

NOTE: In Microsoft Windows XP, the default authentication level is changed from CONNECT to PACKET level. This results in a backward compatibility problem with Windows 2000. Systems running Windows 2000 need to have Service Pack 2 installed to be able to connect to WMI on systems running Windows XP.

Retrieving Dell Information With SMS 2.0

Due to inherent limitations, Microsoft Systems Management Server (SMS) 2.0 can only retrieve information from the **root\cimv2** namespace. OMCI provides information in the **root\dellomci** namespace. However, to make certain key attributes available to SMS, OMCI duplicates them in the **root\cimv2** namespace. For Dell systems running OMCI, you can import the following system attributes into SMS 2.0:

- Service tag
- Asset tag
- System vendor
- System description
- Basic input/output (BIOS) revision
- BIOS date
- Processor type
- Processor speed

To retrieve these Dell attributes, edit the **sms_def.mof** file available on the SMS server in the **sms\inboxes\clifiles.src\hinv** directory. Add the following lines to the end of the **sms_def.mof file**:

```
[SMS_Report(TRUE),
SMS_Group_Name("Dell Summary"),
ResID(7000),ResDLL("dellwmi.dll"),
SMS_Class_ID("Dell|Summary|1.0")]
class Dell_System_Summary : SMS_Class_Template
{
[SMS_Report(TRUE), read: ToInstance ToSubClass]
string SystemVendor;
[SMS_Report(TRUE), read: ToInstance ToSubClass]
string SystemDescription;
[SMS_Report(TRUE), read: ToInstance ToSubClass]
string ProcessorType;
[SMS_Report(TRUE), read: ToInstance ToSubClass]
sint32 ProcessorSpeed;
[SMS_Report(TRUE), read: ToInstance ToSubClass]
string BIOSVersion;
[SMS_Report(TRUE), read: ToInstance ToSubClass]
datetime BIOSDate;
[SMS_Report(TRUE), read: ToInstance ToSubClass, key]
string ServiceTag;
[SMS_Report(TRUE), read: ToInstance ToSubClass,
write: ToInstance ToSubClass]
string AssetTag;
};

#pragma namespace ("\\\\.\\Root\\cimv2")
//**************************************************
**********
//*** Registers Framework Provider ***
//**************************************************
**********
instance of __Win32Provider as $Q
{
Name = "DellWMI";
ClsId = "{a29197b6-5c8d-4278-ae20-e5c14aed1280}";
};
```

```
instance of __InstanceProviderRegistration
{
Provider = $Q;
SupportsGet = TRUE;
SupportsPut = TRUE;
SupportsDelete = TRUE;
SupportsEnumeration = TRUE;
QuerySupportLevels = {"WQL:UnarySelect"};
};
instance of __MethodProviderRegistration
{
Provider = $Q;
};
[Dynamic, Provider ("DellWMI")]
class Dell_System_Summary
{
[read, key] uint32 DellInstanceID;
[read, write] string AssetTag;
[read] datetime BIOSDate;
[read] sint32 ProcessorSpeed;
[read] string BiosVersion;
[read] string ProcessorType;
[read] string ServiceTag;
[read] string SystemDescription;
[read] string SystemVendor;
};
```

WMI Scripts

WMI provides a scripting interface. Using VBScript or JScript, you can write scripts that connect to WMI services locally or remotely, retrieve information, or execute methods. Because OMCI is implemented through WMI, most OMCI tasks can be scripted. This section provides sample VBScript scripts as examples only.

NOTE: To connect remotely to WMI services, you must have administrative rights for both the local and the remote systems.

The following text files contain sample scripts. To execute any of the scripts, save the script with a **.vbs** extension. At the command prompt, type `cscript.exe //nologo <script name>` and the appropriate parameters. If you do not specify any parameters, the usage for the script will be printed out.

NOTICE: The sample scripts are provided as examples only, and have not been tested, nor are warranted in any way by Dell; Dell disclaims any liability in connection therewith. Dell provides no technical support with regard to such scripting. For more information on WMI scripting, see the applicable Microsoft documentation.

- Clear Chassis Intrusion Status
- Disable Desktop Messages for All Alerts
- Remote System Shutdown
- Remote System Restart
- Remote BIOS Update
- Enabling PXE Boot on the Next Reboot
- Enabling Wakeup on LAN
- Retrieving Service Tag, Asset Tag, and BIOS Revision
- Changing BIOS Password

4

Alerts

Overview

Alerts are generated when a system's state of interest changes. There are two basic types of alerting: local alerting to the current computer user and remote alerting to a management application. Dell™ OpenManage™ Client Instrumentation (OMCI) handles both types of alerting.

The Distributed Management Task Force (DMTF) has released the Alert Standard Format (ASF), which defines alerting and remote control interfaces that enable management of networked systems when their operating system is absent. This includes a networked computer that has an inactive or inoperable operating system or that is in a low-power system sleep state. The ASF specification is available at **www.dmtf.org**. ASF documentation is provided with Dell systems that support ASF.

ASF handles remote alerting for environmental events, which are detected by sensors, such as temperature, electrical, and fan probes, and chassis intrusion sensors. ASF also handles additional alerting in the operating-system-absent environment. By default, OMCI handles remote alerting for all events.

ASF 2.0 adds important security measures to the ASF standard, which defines alerting and remote control interfaces to proactively manage networked devices when their operating system is absent. The first industry specification for operating system absent management, ASF allows a network administrator to be alerted to failures of specific components within a networked device, thereby minimizing on-site maintenance while maximizing remote visibility of and access to local systems. Without ASF, operating system absent problems require manual intervention to force a reboot of the system.

There are three settings in the system BIOS that OMCI allows for configuration:

- Off — All ASF 2.0 features are turned off.
- Alert Only — ASF 2.0 alert is the only feature that is turned on.
- On — Alerts and remote control are turned on.

The system network interface card (NIC) and basic input/output system (BIOS) support ASF 2.0. If the BIOS allows for any ASF 2.0 configuration, then the BIOS settings are exposed through the Microsoft® Windows® Management Instrumentation (WMI) by OMCI.

ASF Alerting

In previous versions of OMCI, environmental alerts were sent as Desktop Management Interface (DMI) alerts. With the introduction of ASF, the environmental alerts are handled by the ASF NIC. ASF monitors the environmental probes at a very low level and sends out a Platform Event Trap (PET) when an event occurs. Where applicable, OMCI notifies the local user of the alert. For information on which PET a specific model of Dell system supports, see the system's *Setup Guide*.

ASF is a DMTF management standard that specifies "pre-operating system" or "operating system-absent" alerting techniques. The standard is designed to generate an alert on potential security and fault conditions when the operating system is in a sleep state or the system is powered off. ASF is designed to supersede previous operating system-absent alerting technologies.

WMI Alerting

Local alerting involves displaying user messages and writing to the Windows event log. Remote alerting is accomplished through WMI indications. When OMCI detects an event, it generates an alert, which can be transmitted through the WMI service to a remote management application that is subscribed to that alert type.

When an alert is generated, OMCI supports three types of notification:

- NT event log
- Remote
- Console

In OMCI, each type of event (for example, CurrentProbe, TemperatureProbe, Smart, and so on) that gets logged is provided with an unique event ID number. The events have unique IDs to allow log scraping; this way you can programmatically look at the event log and determine what OMCI events have occurred.

You are also provided with an option to receive either a single alert or a limited number of alerts of the occurrence of an event, of a given type. You can mask out specific events and can generate single alert messages for only those events. The **Dell_IndicationStaticValues** class has the **MaxNTEventLogNotifications**, **MaxDisplayNotifications**, and the **OccurrencesCount** properties, which control the number of notices of each alert condition that are sent to the NT event log and to the local display. For example, setting a value of 0 will not send any notices.

You can change any attribute or CIM class through standard CIM editors. The following properties in the **Dell_IndicationStaticValues** class control the number of notifications of each alert condition that are sent to the NT event log and to the display. You can modify the below listed values through standard CIM editors or through scripts.

- **MaxNTEventLogNotifications:** The maximum number of event notifications sent to the NT event log. Values are 0=never send, 1=first occurrence only, 2=first and second occurrences, and so on. The special value of –1 sends all occurrences. The default value is –1.

- **MaxDisplayNotifications:** The maximum number of event notifications sent to the local display. Values are 0=never send, 1=first occurrence only, 2=first and second occurrences, and so on. The special value of –1 sends all occurrences. The default value is –1.

- **OccurrencesCount:** The number of event notifications of the given alert that have occurred since the last reset. The initial value is 0. Setting **OccurrencesCount** to 0 restarts the sending of notices.

Since the instances of the **Dell_ IndicationStaticValues** class are stored in the WMI repository, the above data is persistent across shutdown, restart, sleep, hibernate, and standby modes. You can reset the **OccurrencesCount** at reboot by installing a script to run at startup.

NOTE: You are not allowed to make changes to some properties including the **MaxDisplayNotifications** and **MaxNTEventLogNotifications** in the **Dell_IndicationStaticValues** class in the CIM repository unless the User Account Control (UAC) in Windows Vista is turned off, or the user is the Administrator account.

The following properties in the **Dell_Configuration** class allows you to control the notices sent to the NT event log and to the local display based on the severity of the event:

- **GlobalNTEventLogSeverityLevel**: Controls the notice to the NT event log based on the severity level of the event

- **GlobalDisplaySeverityLevel**: Controls the notice to the local display based on the severity level of the event

OMCI recognizes the following eight WMI severity levels (represented by integers 0 through 7):

- UNKNOWN = 0
- OTHER = 1
- INFORMATION = 2
- WARNING_DEGRADED = 3
- MINOR = 4
- MAJOR = 5
- CRITICAL = 6
- FATAL_NONRECOVERABLE = 7

NOTE: Events with CRITICAL severity will cause OMCI to shut down the local system after a 60-second delay. The **GlobalDisplaySeverityLevel** property in the **Dell_Configuration** class should always be set to 6 or below; otherwise, the system will not receive a shutdown command from OMCI.

The lowest WMI severity level that OMCI sends is WARNING_DEGRADED and the highest is CRITICAL. The severities of OMCI 7.*x* events are listed in Table 4-1. OMCI sends local alerting and remote alerting for all the listed events.

Table 4-1.   Events Polled by OMCI 7.*x*

OMCI Event (Windows Event Log Number)	Description	Severity Level	Value
FanProbe (1103)	A cooling device has exceeded a minor threshold.	MINOR	4
DiskSizeChanged	The size of at least one hard drive has changed.[1]	WARNING_DEGRADED	3
DiskSizeIncreased (109)	The size of at least one hard drive has increased.[2]	WARNING_DEGRADED	3
DiskSizeDecreased (110)	The size of at least one hard drive has decreased.[2]	WARNING_DEGRADED	3
CurrentProbe (1203)	An electrical current probe has exceeded a minor threshold.	MINOR	4

Table 4-1.   Events Polled by OMCI 7.*x* *(continued)*

OMCI Event (Windows Event Log Number)	Description	Severity Level	Value
DiskCapacity (105)	One of the hard drives is running out of free space.	MINOR	4
	In OMCI 7.0, only one user-defined threshold is used. The threshold is expressed as an absolute value in megabytes (MB). The default value is 50 MB. An event is generated when available free space on one of the hard drives falls below this threshold.		
	In OMCI 7.1 and later versions, two user-defined thresholds are used. An event is generated only when the conditions of both thresholds are met. The first threshold, expressed as an absolute value in megabytes, specifies the size of drives to be monitored. Drives with a capacity smaller than the specified size are ignored. The second threshold is expressed as a percentage of the drive size. An event is generated when available free space on one of the monitored drives falls below this percentage.		
EccMemory (1403)	A memory checksum failure has occurred.	MINOR	4
MemorySizeChanged	The memory size has been changed.[1]	WARNING_DEGR ADED	3

Table 4-1. Events Polled by OMCI 7.*x (continued)*

OMCI Event (Windows Event Log Number)	Description	Severity Level	Value
MemorySizeIncreased (103)	The memory size has been increased.[2]	WARNING_DEGR ADED	3
MemorySizeDecreased (104)	The memory size has been decreased.[2]	WARNING_DEGR ADED	3
NumberOfDisksDecreased (102)	A hard drive has been removed.	WARNING_DEGR ADED	3
NumberOfDisksIncreased (101)	A hard drive has been added.	WARNING_DEGR ADED	3
NumberOfProcessorsDecre ased (107)	A processor has been removed.	WARNING_DEGR ADED	3
NumberOfProcessorsIncre ased (106)	A processor has been added.	WARNING_DEGR ADED	3
Smart (108)	A hard drive condition has occurred that may eventually lead to a drive failure.	MAJOR	5
ChassisIntrusion (1253)	System Chassis Intrusion alert.	WARNING_DEGR ADED	3
TemperatureProbe (1054)	A temperature probe has exceeded a critical threshold.	CRITICAL	6
VoltageProbe (1153)	A voltage probe has exceeded a minor threshold.	MINOR	4

[1] The event is specific to OMCI 7.0.

[2] The event is specific to OMCI 7.1 and later versions.

5

Component Descriptions

Table 5-1 lists the Dell™ instrumentation components, their functions, their default installation locations, and other relevant information.

> **NOTICE:** Changing or deleting component files may cause Dell™ OpenManage™ Client Instrumentation (OMCI) to become unusable.

> **NOTE:** *%ProgramFiles%* is a variable that refers to the path to the Program Files directory. By default, this is **C:\Program Files**.

Table 5-1.   Instrumentation Component Information

Component	Default Location or File Name	Functional Information
CIM Provider	%ProgramFiles%\Dell\Open Manage\ Client\DellWMI.dll	Module that provides access to information requested by the Microsoft® Windows® Management Instrumentation (WMI) Service Layer; retrieves data for the attributes specified by the Managed Object Format (MOF), and exposes data through the Common Information Model Object Manager (CIMOM).
		Most of its functionality is devoted to provided class instances; however, it also implements three WMI methods: client shutdown, restart and BIOS flash.
Disks	%ProgramFiles%\Dell\Open Manage\ Client\DDisks.dll	Data-provider dynamic link library (DLL) that provides information about Integrated Drive Electronics (IDE) and Small Computer System Interface (SCSI) Self-Monitoring Analysis and Reporting Technology (SMART) drives.

Table 5-1. Instrumentation Component Information *(continued)*

Component	Default Location or File Name	Functional Information
SMBIOS	%ProgramFiles%\Dell\Open Manage\ Client\dsmbios.dll	Component that retrieves data from the system management basic input/output system (SMBIOS) data tables and reads and writes Dell system data into the BIOS through system management interrupts (SMIs) and the CMOS.
IAP	%ProgramFiles%\Dell\Open Manage\ Client\Iap.exe	Instrumentation Access Provider (IAP); component that queries the **iap_routing_data.htm** file for configuration information, builds the lookup tables, and routes data to a data provider; ensures serialization of BIOS access and consistency of data, and reduces memory and disk requirements.
		IAP is an executable that runs as a Windows service and is the only service installed by OMCI. Its purpose is to route requests from the layer above to the appropriate component provider below. It also hosts the component DLLs (**dsmbios.dll**, **dpci.dll**, or **dsysInfo.dll**) while they are in use.
WDM Driver	**C:\WINNT\system32\drivers\ omci.sys** (for Microsoft Windows XP Professional) **C:\Windows\System32\drivers \omci.sys** (for Microsoft Windows Vista®)	Windows Driver Model (WDM) driver that provides various components with access to hardware.

Table 5-1. Instrumentation Component Information *(continued)*

Component	Default Location or File Name	Functional Information
PC Card	**%ProgramFiles%\Dell\Open Manage\ Client\dpccard.dll**	Data-provider DLL that gives information about the PC Cards (removable I/O cards such as a modem, local area network [LAN], synchronous RAM [SRAM], or flash memory card) present in a system.
PCI	**%ProgramFiles%\Dell\Open Manage\ Client\dpci.dll**	Data-provider DLL that gives information about Peripheral Component Interconnect (PCI) devices, PCI-PCI bridges, and PCI-Card bus bridges present in a system.
pcidb.txt	**%ProgramFiles%\Dell\Open Manage\Client**	Lists most of the PCI devices in the industry.
dcpcidev.txt	**%ProgramFiles%\Dell\Open Manage\Client**	Dell specific PCI device list.
System Information	**%ProgramFiles%\Dell\Open Manage\ Client\dsysInfo.dll**	Data-provider DLL that provides system-specific information not available through other data-provider DLLs.
Events DLL	**%ProgramFiles%\Dell\Open Manage\ Client\DEvents.dll**	DLL that has a collection of Component Object Model (COM) Objects, each assigned to a specific event. The event modules communicate with IAP and other components, for information about the system to determine if the user or administrator needs notification.
Monitor	**%ProgramFiles%\Dell\Open Manage\ Client\dmonitor.dll**	Data-provider DLL that provides information about the monitor(s) attached to the system.

Table 5-1.   Instrumentation Component Information *(continued)*

Component	Default Location or File Name	Functional Information
Indication Consumer	%ProgramFiles%\Dell\Open Manage\ Client\IndiCons.dll	DLL that subscribes to the CIMOM for indications of interest. It is supplied by the indication provider when events are detected. The providers send indications to the CIMOM, and the CIMOM forwards them to their subscribers.
		Indication consumer can interpret an indication and perform certain actions in response to the event, such as reporting the event to the user, executing a script, or shutting down the system.
Indication Provider	%ProgramFiles%\Dell\Open Manage\ Client\IndiProv.dll	DLL that periodically polls for events occurring at multiple event sources and generates an indication when an event is detected. It then supplies the indication to WMI, which forwards it to indication consumers (subscribers). The Indication Provider detects events by calling the COM interfaces of event source status checkers at configurable intervals. These status checkers are all implemented in **DEvents.dll**. The status checkers call the IAP for data, and detect events by examining the returned data. For each event type, there is a corresponding status checker.
CIM-IAP Data Routing File	%ProgramFiles%\Dell\Open Manage\ Client\cim_iap_data.htm	HTML file that contains configuration information for **DellWMI.dll**.
CIM-Win32 Data Routing File	%ProgramFiles%\Dell\Open Manage\ Client\cim_win32_data.htm	HTML file that contains configuration information for **DellWMI.dll**.

Table 5-1. Instrumentation Component Information *(continued)*

Component	Default Location or File Name	Functional Information
DSMBIOS Data Routing File	**%ProgramFiles%\Dell\Open Manage\ Client\dsmbios_data.htm**	HTML file that contains configuration information for **dsmbios.dll**.
IAP Data Routing File	**%ProgramFiles%\Dell\Open Manage\ Client\iap_routing_data.htm**	HTML file that contains configuration information for **iap.exe**.
Dell OMCI Namespace Create MOF[1]	**%ProgramFiles%\Dell\Open Manage\ Client\Mofs\ DellOMCI_NamespaceCreat e.mof**	MOF file that creates the **\root\dellomci** namespace on the client systems.
Dell OMCI Windows Preparation MOF[2]	**%ProgramFiles%\Dell\Open Manage\ Client\Mofs\DellOMCI_Win Prep.mof**	MOF file that creates the **\root\dellomci** namespace on the client systems and also initializes the alerting.
CIM v26 LT MOF	**%ProgramFiles%\Dell\Open Manage\ Client\Mofs\cim26lt.mof**	MOF file that includes all parent classes of **DellOMCI.mof**. These classes are from Distributed Management Task Force (DMTF) standard CIM Schema Version 2.6.
Dell OMCI MOF	**%ProgramFiles%\Dell\Open Manage\ Client\Mofs\DellOMCI.mof**	MOF file that defines all the CIM classes of OMCI.
Dell OMCI Qualifier MOF[2]	**%ProgramFiles%\Dell\Open Manage\ Client\Mofs\DellOMCI_Qua lifiers.mof**	MOF file that contains declarations of the qualifiers used in OMCI, but which are not part of the CIM standard.
Dell OMCI Schema MOF[2]	**%ProgramFiles%\Dell\Open Manage\ Client\Mofs\DellOMCI_Sch ema.mof**	MOF file that contains **#pragma** include statements for each of the MOF files in OMCI.
Dell COO Data MOF[1]	**%ProgramFiles%\Dell\Open Manage\ Client\Mofs\coodata.mof**	MOF file that provides the default values to the instances of the Cost of Ownership (COO) classes defined in **DellOMCI.mof**.

Table 5-1.   Instrumentation Component Information *(continued)*

Component	Default Location or File Name	Functional Information
Dell Instantiation MOF[2]	%ProgramFiles%\Dell\Open Manage\ Client\Mofs\DellOMCI_Inst ant.mof	MOF file that provides the default values to the instances of the Cost of Ownership (COO) classes defined in **DellOMCI.mof**.
CIM v26 LT MFL[1]	%ProgramFiles%\Dell\Open Manage\ Client\Mofs\cim26lt.mfl	Amended MOF file from **cim26lt.mof**. The corresponding translated version of this file is loaded and compiled according to the selected installation language.
Dell OMCI MFL[1]	%ProgramFiles%\Dell\Open Manage\ Client\Mofs\dellomci.mfl (Installed only for non-English OMCI installations)	Amended MOF file from **dellomci.mof**. The corresponding translated version of this file is loaded and compiled according to the selected installation language.
Dell Event MOF[1]	%ProgramFiles%\Dell\Open Manage\ Client\Mofs\dellevent.mof	MOF file that contains the language-neutral part of the classes and instances of the OMCI 7.0 Event MOF file. This file should be loaded and compiled on all the client systems regardless of the language.
Dell Event MFL[1]	%ProgramFiles%\Dell\Open Manage\ Client\Mofs\dellevent.mfl	Amended MOF file that contains the translated strings of the OMCI 7.0 Event MOF file. This file should be loaded and compiled according to the selected installation language.
Dell Event Base MFL[1]	%ProgramFiles%\Dell\Open Manage\ Client\Mofs\debase.mfl	Amended MOF file that contains the English version of the base string of the OMCI 7.0 Event MOF file. This file should be loaded and compiled on all the client systems regardless of the language.

[1] The component is specific to OMCI 7.0.

[2] The component is specific to OMCI 7.1 and later.

6

Troubleshooting

Wakeup on LAN

If Wakeup on LAN does not function on a client computer system, perform the following checks:

- Use either Dell™ OpenManage™ IT Assistant or system setup on the client system to verify that Wakeup on LAN is enabled. See your Dell system documentation for detailed information.

- Ensure that the correct Network Interface Card (NIC) driver is installed. Updated drivers are available on the Dell support website at **support.dell.com**.

- Perform the following steps:

1 In **Device Manager**, double-click the NIC to open the **NIC properties** window.

2 Click **Configure**.

3 On the **Advanced** tab, set the **Wake-up** property to **Magic Packet.**

Remote Connection to WMI and Remote BIOS Update

If Common Information Model (CIM) information for a remote client computer system is not available to the management application, or if remote BIOS update, which uses Distributed Component Object Model (DCOM), fails, `Access Denied` and/or `Win32:RPC server is unavailable` error messages may be displayed. If you encounter these error messages, perform the following troubleshooting steps:

- To verify that the client system is connected to the network, at the command prompt on the server, type:

  `ping <Host Name or IP Address>` and press <Enter>.

- If both the server and the client system are in a domain, perform the following steps:
  - Verify that the domain administrator account has administrator privileges for both systems.
  - If you are using IT Assistant on the server system, run the IT Assistant ConfigServices utility (**configservices.exe** found in the **/bin** directory of the IT Assistant installation directory). Configure IT Assistant to run under the account of the domain administrator, and verify that DCOM and CIM are enabled. See the *IT Assistant User's Guide* for information.
  - If you are using IT Assistant, use the domain administrator account to configure the IT Assistant subnet discovery for the client system. If the system has already been discovered, remove the system from the list of discovered systems, configure subnet discovery for it, and then rediscover it. See the *IT Assistant User's Guide* for information.
- If both the server and the client system are in a workgroup (not in a domain), perform the following steps:
  - If the client system is running the Microsoft® Windows® XP Professional or Windows XP Professional x64 Edition operating system and the server is running Windows 2000, verify that Service Pack 2 or later is installed on the server.
  - If the client system is running a version of Windows XP and is not part of a domain, edit the registry to allow remote connection with administrative privileges.

    ⊘ **NOTICE:** Back up your system data files before making changes to the registry. Incorrectly editing the registry may render your operating system unusable.

    To make the registry change on the client system, click **Start**→**Run**, then type regedit, and click **OK**. In the **Registry Editor** window, navigate to:

    ```
    My Computer\HKEY_LOCAL_MACHINE\SYSTEM\
    ```

    ```
    CurrentControlSet\Control\Lsa
    ```

    Set the **forceguest** value to **0** (by default it is **1**). Unless you modify this value, the user remotely connecting to the system has guest privileges only, even if the supplied credentials should provide administrative privileges.

– Create an account on the client system with the same user name and password as an administrator account on the system running your WMI management application.

– If you are using IT Assistant, run the IT Assistant ConfigServices utility (**configservices.exe** found in the **/bin** directory under the IT Assistant installation directory). Configure IT Assistant to run under a local administrator account, which is also now an administrator on the remote client. Also, verify that DCOM and CIM are enabled. See the *IT Assistant User's Guide* for information.

– If you are using IT Assistant, use the administrator account to configure subnet discovery for the client system. Enter the user name as `<client machine name>\<account name>`. If the system has already been discovered, remove the system from the list of discovered systems, configure subnet discovery for it, and then rediscover it. See the *IT Assistant User's Guide* for information.

- Perform the following steps in order to modify user privilege levels for connecting remotely to a system's WMI:

  – Click **Start**, click **Run**, type **compmgmt.msc**, and then click **OK**.

  – Navigate to **WMI Control** under **Services and Applications**.

  – Right click **WMI Control** and then click **Properties**.

  – Click the **Security** tab and then select **DellOMCI** under the **Root** tree.

  – Click the **Security** button.

  – Select the specific group or user that you want to control access and use the **Allow** or **Deny** checkbox to configure permissions.

- Perform the following steps to connect to a system's WMI (**root\dellomci**) from a remote system using WMI CIM Studio:

  – Install **WMI tools** along with **CIM Studio** on the local system and install **Dell OMCI** on the remote system.

  – Configure the system's firewall accordingly for WMI Remote Connectivity. For example, open the TCP ports 135 and 445 in Windows Firewall.

– Set the **Local Security** setting to `Classic - local users authenticate as themselves` for `Network access: Sharing and security model for local accounts` in the Local Security Policy.

– In case of Windows Vista®, disable UAC on the system.

– Now connect to the system's WMI (**root\dellomci**) from a remote system using WMI CIM Studio (for example, **\\10.94.174.167\root\dellomci**)

– Enter the Administrator credentials of the target remote system if prompted.

NOTE: See the applicable Microsoft documentation at **http://msdn.microsoft.com** for more information.

Installation Problems

If you are unable to complete installation of OMCI, check the following:

• You must have administrative rights on the target system.

• The target system must be a Dell system with SMBIOS version 2.3 or later.

• The system must be running the Microsoft Windows operating systems listed in the "Supported Operating Systems" section.

A

Sample Scripts and BIOS Settings

WMI Sample Scripts

Clear Chassis Intrusion Status

Below is a sample VBScript that will clear the chassis intrusion status of a system. This script can be run locally or remotely using the name of the system.

```
Namespace: root\DellOMCI

Class Name: Dell_SMBIOSsettings

Instance 'Key' Value: 0

Property Name: ChassisIntrusionStatus

Property Value: 5
'**************************************************
'*** Name: SampleChassisIntrusionClear.vbs

'*** Purpose: To clear the chassis intrusion status on
a Dell OMCI client.

'*** Usage: cscript.exe //nologo
SampleChassisIntrusionClear.vbs <systemname>

'***

'*** This sample script is provided as an example
'***only, and has not been tested, nor is
'***warranted in any way by Dell; Dell disclaims any
'*** liability in connection therewith. Dell provides
'***no technical support with regard to such
'***scripting. For more information on WMI
'***scripting, refer to applicable Microsoft
'***documentation.
'**************************************************
Option Explicit
```

```
'*** Declare variables

Dim strNameSpace

Dim strComputerName

Dim strClassName

Dim strKeyValue

Dim objInstance

Dim strPropName

Dim strPropValue

'*** Check that the right executable was used to run
'***the script and that all parameters were passed.

If (LCase(Right(WScript.FullName, 11)) =
"wscript.exe" ) Or _ (Wscript.Arguments.Count < 1)
Then

Call Usage()

WScript.Quit

End If

'*** Initialize variables

strNameSpace = "root/Dellomci"

strComputerName = WScript.Arguments(0)

strClassName = "Dell_SMBIOSsettings"

strKeyValue = "0"

strPropName = "ChassisIntrusionStatus"

'*** Retrieve the instance of Dell_SMBIOSSettings
'***class (there should only be 1 instance).

Set objInstance =
GetObject("WinMgmts:{impersonationLevel=impersonate
,AuthenticationLevel=pktprivacy }//" &_
strComputerName & "/" & strNameSpace & ":" &
strClassName & "=" &_ Chr(34) & strKeyValue & Chr(34))
```

```
'*** Set the value of ChassisIntrusionStatus to'5'
'***("Clear")

'***Set the new value for the property and save the
instance
objInstance.Properties_.Item(strPropName).Value = 5
objInstance.Put_

'*** If any errors occurred, let the user know.

If Err.Number <> 0

Then WScript.Echo "Clearing Chassis Intrusion Status
failed."

End If

'*** Sub used to display the correct usage of the
'***script

Sub Usage()

Dim strMessage

strMessage = "incorrect syntax. You should run: " &
vbCRLF & _ "cscript.exe /nologo
SampleChassisIntrusionClear.vbs <systemname>"
WScript.Echo strMessage

End Sub
```

Disable Desktop Messages for All Alerts

Below is a sample VBScript that will disable the message box that is displayed on the desktop if an alert condition is detected.

```
Namespace: root\DellOMCI

Class Name: Dell_IndicationStaticValues

Instance 'Key' Value: Not Applicable to this example

Property Name: MaxDisplayNotifications

Property Value: 0
'*****************************************************
'*** Name: SampleDisableAlertMessages.vbs
```

```
'*** Purpose: To disable the desktop message box for
'***all alerts on a Dell OMCI client.

'***Usage: cscript.exe //nologo
'***SampleDisableAlertMessages.vbs <systemname>'***

'*** This sample script is provided as an example
'***only, and has not been tested, nor is warranted in
'***any way by Dell; Dell disclaims any liability in
'***connection therewith. Dell provides no technical
'***support with regard to such scripting. For more
'***information on WMI scripting, refer to applicable
'***Microsoft documentation.
'****************************************************
Option Explicit

'*** Declare variables

Dim objWMIService

Dim strNameSpace

Dim strComputerName

Dim strClassName

Dim ColSystem

Dim objInstance

Dim strPropName

Dim strPropValue

'*** Check that the right executable was used to run
'***the script and that all parameters were passed

If (LCase(Right(WScript.FullName, 11)) =
"wscript.exe" ) Or _

(Wscript.Arguments.Count < 1) Then

Call Usage()

WScript.Quit

End If
```

```
'*** Initialize variables

strNameSpace = "root/Dellomci"

strComputerName = WScript.Arguments(0)

strClassName = "Dell_IndicationStaticValues"

strPropName = "MaxDisplayNotifications"

'*** Establish a connection to the DellOMCI namespace
Set objWMIService =
GetObject("winmgmts:{impersonationLevel=impersonate,"
&_ "AuthenticationLevel=pktprivacy}\\" &
strComputerName & "\" &_ strNameSpace)

'*** Retrieve the instances of
'***Dell_IndicationStaticValues class (there should
be 16 instances – one for each alert type).

Set ColSystem=objWMIService.execquery ("Select * from
" & strClassName)

For each objInstance in ColSystem

'*** Set the value of MaxDisplayNotifications to'0'
'*** Set the new value for the property and save the
instance
objInstance.Properties_.Item(strPropName).Value = 0
objInstance.Put_ Next

'*** If any errors occurred, let the user know.

If Err.Number <> 0 Then

WScript.Echo "The change to the property failed."

End If

'*** Sub used to display the correct usage of the
'***script

Sub Usage()

Dim strMessage
```

```
strMessage = "incorrect syntax. You should run: " &
vbCRLF & _ "cscript.exe /nologo
SampleDisableAlertMessages.vbs <systemname>"
WScript.Echo strMessage

End Sub
```

Remote System Shutdown

Below is a sample VBScript that will shut down a Dell™ OMCI client
remotely.

```
'***************************************************
'*** Name: SampleShutdown.vbs
'*** Purpose: To shut down a Dell™ OMCI client.
'*** Usage: cscript.exe //nologo SampleShutdown.vbs
<systemname>
'***
'*** This sample script is provided as an example
only, and has not
'*** been tested, nor is warranted in any way by Dell;
Dell disclaims
'*** any liability in connection therewith. Dell
provides no
'*** technical support with regard to such scripting.
For more
'*** information on WMI scripting, refer to applicable
Microsoft®
'*** documentation.
'***************************************************
Option Explicit

'*** Declare variables
Dim strNameSpace
Dim strComputerName
Dim strClassName
Dim strPropValue
Dim objInstance
Dim ObjOutParam
Dim strMethod
```

```
'*** Check that the right executable was used to run
the script
'*** and that all parameters were passed
If (LCase(Right(WScript.FullName, 11)) =
"wscript.exe" ) Or _
        (Wscript.Arguments.Count < 1) Then
        Call Usage()
        WScript.Quit
End If
'*** Initialize variables
strNameSpace = "root/Dellomci"
strComputerName = WScript.Arguments(0)
strClassName = "Dell_Configuration"
strPropValue = "Configuration"
strMethod = "Shutdown"

'*** Retrieve the Dell_Configuration class
Set objInstance =
GetObject("WinMgmts:{impersonationLevel=
impersonate}//" &_
        strComputerName & "/" & strNameSpace & ":" &
strClassName)

'*** Execute the Shutdown method
Set ObjOutParam = objInstance.ExecMethod_(strMethod)

'*** Let the user know whether the method returned
success or not
'*** Note: if return value is 0, it does not mean the
system has
'*** shutdown, it only means that the shutdown method
was initiated
'*** successfully.
If objOutParam.ReturnValue = 0 Then
        WScript.Echo "Method completed successfully."
else
        WScript.Echo "Method failed."
End If
```

```
'*** Sub used to display the correct usage of the
script
Sub Usage()
Dim strMessage
strMessage = "incorrect syntax. You should run: " &
vbCRLF & _
        "cscript.exe //nologo SampleShutdown.vbs
<systemname>"
WScript.Echo strMessage
End Sub
```

Remote System Restart

Below is a sample VBScript that will restart a Dell OMCI client remotely.

```
'*************************************************
'*** Name: SampleRestart.vbs
'*** Purpose: To restart a Dell OMCI client.
'*** Usage: cscript.exe //nologo SampleRestart.vbs
<systemname>
'***
'*** This sample script is provided as an example
only, and has not been
'*** tested, nor is warranted in any way by Dell; Dell
disclaims any
'*** liability in connection therewith. Dell provides
no technical
'*** support with regard to such scripting. For more
information on WMI
'*** scripting, refer to applicable Microsoft
documentation.
'*************************************************

Option Explicit

'*** Declare variables
Dim strNameSpace
Dim strComputerName
Dim strClassName
Dim strPropValue
```

```
Dim objInstance
Dim ObjOutParam
Dim strMethod

'*** Check that the right executable was used to run
the script
'*** and that all parameters were passed
If (LCase(Right(WScript.FullName, 11)) =
"wscript.exe" ) Or _
        (Wscript.Arguments.Count < 1) Then
        Call Usage()
        WScript.Quit
End If

'*** Initialize variables
strNameSpace = "root/Dellomci"
strComputerName = WScript.Arguments(0)
strClassName = "Dell_Configuration"
strPropValue = "Configuration"
strMethod = "Restart"

'*** Retrieve the Dell_Configuration class
Set objInstance =
GetObject("WinMgmts:{impersonationLevel=
impersonate}//" &_
        strComputerName & "/" & strNameSpace & ":" &
strClassName)

'*** Execute the Shutdown method
Set ObjOutParam = objInstance.ExecMethod_(strMethod)

'*** Let the user know whether the method returned
success or not
'*** Note: if return value is 0, it does not mean the
system has
'*** shutdown, it only means that the shutdown method
was initiated
'*** successfully.
If objOutParam.ReturnValue = 0 Then
        WScript.Echo "Method completed successfully."
else
```

```
        WScript.Echo "Method failed."
End If

'*** Sub used to display the correct usage of the
script
Sub Usage()
Dim strMessage
strMessage = "incorrect syntax. You should run: " &
vbCRLF & _
        "cscript.exe //nologo SampleRestart.vbs
<systemname>"
WScript.Echo strMessage
End Sub
```

Remote BIOS Update

Below is a sample VBScript that will flash the BIOS of a Dell OMCI client
remotely.

```
'****************************************************
'*** Name: SampleFlash.vbs
'*** Purpose: To flash the BIOS of a Dell OMCI client.
'*** Usage: cscript.exe //nologo SampleFlash.vbs
<systemname> <URL
'*** of BIOS header file>
'***
'*** This sample script is provided as an example
only, and has not
'*** been tested, nor is warranted in any way by Dell;
Dell disclaims any
'*** liability in connection therewith.Dell provides
no technical
'*** support with regard to such scripting. For more
information on
'*** WMI scripting, refer to applicable Microsoft
documentation.
'****************************************************

Option Explicit
```

```
'*** Declare variables
Dim strNameSpace
Dim strComputerName
Dim strClassName
Dim strPropValue
Dim objInstance
Dim ObjOutParam
Dim strMethod
Dim objMethod
Dim objClass
Dim objInParam


'*** Check that the right executable was used to run
the script
'*** and that all parameters were passed
If (LCase(Right(WScript.FullName, 11)) =
"wscript.exe" ) Or _
        (Wscript.Arguments.Count < 2) Then
        Call Usage()
        WScript.Quit
End If

'*** Initialize variables
strNameSpace = "root/Dellomci"
strComputerName = WScript.Arguments(0)
strClassName = "Dell_Configuration"
strPropValue = "Configuration"
strMethod = "FlashBios"

'*** Retrieve the Dell_Configuration class
Set objClass =
GetObject("WinMgmts:{impersonationLevel=
impersonate}//" & _
        strComputerName & "/" & strNameSpace & ":" &
strClassName)

Set objMethod = objClass.Methods_(strMethod)
```

```
'*** Set the In parameter of the method to the URL of
BIOS header file
Set objInParam =
objMethod.inParameters.SpawnInstance_()
objInParam.sUrl = WScript.Arguments(1)

'*** Execute the method
Set ObjOutParam = objClass.ExecMethod_(strMethod,
objInParam)

'*** Let the user know whether the method returned
success or not
'*** Note: if return value is 0, it does not mean the
system's BIOS has
'*** been flashed, it only means that the flash method
was initiated
'*** successfully.
If objOutParam.ReturnValue = 0 Then
        WScript.Echo "Method completed successfully."
else
        WScript.Echo "Method failed."
End If


'*** Sub used to display the correct usage of the
script
Sub Usage()
Dim strMessage
strMessage = "incorrect syntax. You should run: " &
vbCRLF & _
        "cscript.exe //nologo SampleFlash.vbs
<systemname> <URL of BIOS
        header file>"
WScript.Echo strMessage
End Sub
```

Enabling PXE Boot on the Next Reboot

Below is a sample VBScript that will force a Dell OMCI client to boot to PXE on next reboot.

```
'***************************************************
'*** Name: SampleForcePXE.vbs
'*** Purpose: To force a Dell OMCI client to boot to
PXE on next reboot.
'*** Usage: cscript.exe //nologo SampleForcePXE.vbs
<systemname>
'***
'*** This sample script is provided as an example
only, and has not been
'*** tested, nor is warranted in any way by Dell; Dell
disclaims any
'*** liability in connection therewith. Dell provides
no technical
'*** support with regard to such scripting. For more
information on WMI
'*** scripting, refer to applicable Microsoft
documentation.
'***************************************************

Option Explicit

'*** Declare variables
Dim strNameSpace
Dim strComputerName
Dim strClassName
Dim strKeyValue
Dim objInstance
Dim strPropName
Dim strPropValue

'*** Check that the right executable was used to run
the script
'*** and that all parameters were passed
If (LCase(Right(WScript.FullName, 11)) =
"wscript.exe" ) Or _
```

```
        (Wscript.Arguments.Count < 1) Then
        Call Usage()
        WScript.Quit
End If

'*** Initialize variables
strNameSpace = "root/Dellomci"
strComputerName = WScript.Arguments(0)
strClassName = "Dell_Configuration"
strKeyValue = "Configuration"
strPropName = "ForcePXEOnNextBoot"

'*** Retrieve the instance of Dell_Configuration class
(there should
'*** only be 1 instance).
Set objInstance =
GetObject("WinMgmts:{impersonationLevel=
impersonate}//" &_
        strComputerName & "/" & strNameSpace & ":" &
strClassName & "=" & _
        Chr(34) & strKeyValue & Chr(34))


strPropValue =
objInstance.Properties_.Item(strPropName).Value

'*** Set the new value for the property and save the
instance, but only
'*** if the current value is not already 3 ('Enabled')
If strPropValue <> 3 Then
        objInstance.Properties_.Item(strPropName).Val
ue = 3
        objInstance.Put_

        '*** If any errors occurred, let the user know
        If Err.Number <> 0 Then
                WScript.Echo "Setting PXE on next
reboot failed."
        End If
End If
```

```
'*** Sub used to display the correct usage of the
script
Sub Usage()
Dim strMessage
strMessage = "incorrect syntax. You should run: " &
vbCRLF & _
        "cscript.exe //nologo SampleForcePXE.vbs
<systemname>"
WScript.Echo strMessage
End Sub
```

Enabling Wakeup on LAN

Below is a sample VBScript that will enable Wakeup On LAN on a Dell
OMCI client.

```
'**************************************************
'*** Name: SampleWuOLEnable.vbs
'*** Purpose: To enable Wakeup On LAN on a Dell OMCI
client.
'*** Usage: cscript.exe //nologo SampleWuOLEnable.vbs
<systemname>
'***
'*** This sample script is provided as an example
only, and has not been
'*** tested, nor is warranted in any way by Dell; Dell
disclaims any
'*** liability in connection therewith. Dell provides
no technical
'*** support with regard to such scripting. For more
information on WMI
'*** scripting, refer to applicable Microsoft
documentation.
'**************************************************


Option Explicit

'*** Declare variables
Dim strNameSpace
```

```
Dim strComputerName
Dim strClassName
Dim strKeyValue
Dim objInstance
Dim strPropName
Dim strPropValue

'*** Check that the right executable was used to run
the script
'*** and that all parameters were passed
If (LCase(Right(WScript.FullName, 11)) =
"wscript.exe" ) Or _
        (Wscript.Arguments.Count < 1) Then
        Call Usage()
        WScript.Quit

End If
'*** Initialize variables
strNameSpace = "root/Dellomci"
strComputerName = WScript.Arguments(0)
strClassName = "Dell_SMBIOSSettings"
strKeyValue = "0"
strPropName = "WakeupOnLan"
'*** Retrieve the instance of Dell_SMBIOSSettings
class (there should
'*** only be 1 instance).
Set objInstance =
GetObject("WinMgmts:{impersonationLevel=
impersonate}//" &_
        strComputerName & "/" & strNameSpace & ":" &
strClassName & "=" & _
        Chr(34) & strKeyValue & Chr(34))

strPropValue =
objInstance.Properties_.Item(strPropName).Value

'*** Set the value of WakeUpOnLan only if it is not
already '6' ("Enable
'*** for all NICs")
if strPropValue <> 6 then
```

```
        '*** Set the new value for the property and
save the instance
        objInstance.Properties_.Item(strPropName).Val
ue = 6
        objInstance.Put_

        '*** If any errors occurred, let the user know
        If Err.Number <> 0 Then
                WScript.Echo "Enabling WakeUp On Lan
failed."
        End If
End If

'*** Sub used to display the correct usage of the
script
Sub Usage()
Dim strMessage
strMessage = "incorrect syntax. You should run: " &
vbCRLF & _
        "cscript.exe //nologo SampleWuOLEnable.vbs
<systemname>"
WScript.Echo strMessage
End Sub
```

Retrieving Service Tag, Asset Tag, and BIOS Revision

Below is a sample VBScript that will display the Asset Tag, Service Tag, and
the BIOS revision of a Dell OMCI client.

```
'**************************************************
'*** Name: SampleSystemSummary.vbs
'*** Purpose: To display asset tag, service tag, and
BIOS revision of
'*** a Dell OMCI client.
'*** Usage: cscript.exe //nologo
SampleSystemSummary.vbs <systemname>
'***
'*** This sample script is provided as an example
only, and has not been
'*** tested, nor is warranted in any way by Dell; Dell
```

```
disclaims any
'*** liability in connection therewith. Dell provides
no technical
'*** support with regard to such scripting. For more
information on WMI
'*** scripting, refer to applicable Microsoft
documentation.
'**************************************************

Option Explicit

'*** Declare variables
Dim strNameSpace
Dim strComputerName
Dim strClassName
Dim colInstances
Dim objInstance
Dim strWQLQuery
Dim strMessage
Dim strKeyName

'*** Check that the right executable was used to run
the script
'*** and that all parameters were passed
If (LCase(Right(WScript.FullName, 11)) =
"wscript.exe" ) Or _
        (Wscript.Arguments.Count < 1) Then
        Call Usage()
        WScript.Quit
End If

'*** Initialize variables
strNameSpace = "root/Dellomci"
strComputerName = WScript.Arguments(0)
strClassName = "Dell_SystemSummary"
strKeyName = "Name"

'*** WQL Query to retrieve instances of
Dell_SystemSummary
strWQLQuery = "SELECT * FROM " & strClassName & "
```

```
WHERE " & _
        strKeyName & "=" & Chr(34) & strComputerName &
Chr(34)

'*** Retrieve instances of Dell_Configuration class
(there should only
'*** be 1 instance).
Set colInstances =
GetObject("WinMgmts:{impersonationLevel=
impersonate}//"&_
        strComputerName & "/" &
strNameSpace).ExecQuery(strWQLQuery, "WQL",
        NULL)

'*** Use only first instance to retrieve asset tag,
service tag, and BIOS
'*** version
For Each objInstance in colInstances
        strMessage = "Asset Tag: "
        strMessage = strMessage &
objInstance.Properties_.Item
        ("AssetTag").Value
        strMessage = strMessage & vbCRLF & "Service
Tag: "
        strMessage = strMessage &
objInstance.Properties_.Item
        ("ServiceTag").Value
        strMessage = strMessage & vbCRLF & "BIOS
Version: "
        strMessage = strMessage &
objInstance.Properties_.Item
        ("BIOSVersion").Value
    Exit For
Next

'*** Display the results
WScript.Echo strMessage

'*** Sub used to display the correct usage of the
script
```

```
Sub Usage()
Dim strMessage
strMessage = "incorrect syntax. You should run: " &
vbCRLF & _
        "cscript.exe //nologo SampleSystemSummary.vbs
<systemname>"
WScript.Echo strMessage
End Sub
```

Changing BIOS Password

Below is a sample VBScript that will change the BIOS password on a Dell OMCI client.

```
'**************************************************
'*** Name: SampleBIOSPwd.vbs
'*** Purpose: To change the BIOS password on a Dell
OMCI client.
'*** Usage: cscript.exe //nologo SampleBIOSPwd.vbs
<systemname> "<old
'*** pwd> space <new pwd>"
'***
'*** This sample script is provided as an example
only, and has not been
'*** tested, nor is warranted in any way by Dell; Dell
disclaims any
'*** liability in connection therewith. Dell provides
no technical
'*** support with regard to such scripting. For more
information on WMI
'*** scripting, refer to applicable Microsoft
documentation.
'**************************************************

'*** Declare variables
Dim strNameSpace
Dim strComputerName
Dim strClassName
Dim strKeyValue
Dim objInstance
```

```
Dim strPropName
Dim strPwd

'*** Check that the right executable was used to run
the script
'*** and that all parameters were passed
If (LCase(Right(WScript.FullName, 11)) <>
"cscript.exe" ) Or _
        (Wscript.Arguments.Count < 2) Then
        Call Usage()
        WScript.Quit
End If
'*** Initialize variables
strNameSpace = "root/Dellomci"
strComputerName = WScript.Arguments(0)
strClassName = "Dell_Configuration"
strKeyValue = "Configuration"
strPropName = "Password"
strPassEncryptPropName = "PasswordEncrypted"
strPwd = WScript.Arguments(1)

'*** Retrieve the instance of Dell_Configuration class
(there should
'*** only be 1 instance).
Set objInstance =
GetObject("WinMgmts:{impersonationLevel=
impersonate}//" &_
        strComputerName & "/" & strNameSpace & ":" &
strClassName & "=" & _
        Chr(34) & strKeyValue & Chr(34))

'*** Set the new value for the property and save the
instance
objInstance.Properties_.Item(strPropName).Value =
strPwd
objInstance.Properties_.Item(strPassEncryptPropName).
Value = 0
objInstance.Put_
```

```
'*** If any errors occurred, let the user know
If Err.Number <> 0 Then
        WScript.Echo "Setting the BIOS password
failed."
End If

'*** Sub used to display the correct usage of the
script
Sub Usage()
Dim strMessage
strMessage = "Incorrect syntax. You should run: " &
vbCRLF & _
        "cscript.exe //nologo SampleBIOSPwd.vbs
<systemname> " & Chr(34) & _
        "<old pwd> space <new pwd>" & Chr(34)
WScript.Echo strMessage
End Sub
```

BIOS Settings Supported in OMCI 7.6

NOTE: Not all of the below listed BIOS settings are supported on all Dell systems.

The BIOS settings supported in OMCI 7.6 are:

- AmbientLightSensor*
- AGPSlot*
- ASFMode
- AudioMode
- AutoOn
- AutoOnHour
- AutoOnMinute
- BluetoothDevices*
- BootSequence
- BuiltinFloppy
- BuiltinNIC
- BuiltinPointingDevice

- CellularRadio
- ChassisIntrusion
- ChassisIntrusionStatus
- ClearSystemEventLog*
- Cool and Quiet*
- CPUVirtualization*
- DASH On*
- Embedded RAID Controller*
- Embedded NIC2*
- E-SATA Ports*
- ExternalHotkey*
- FastBoot
- HardDiskFailover*
- HotDocking
- Hyperthreading

✎ **NOTE:** On some Dell systems, the **Hyperthreading** property in the
Dell_SMBIOSSettings class displays **Enabled** or **Disabled** even if the installed
processor does not support hyperthreading technology. In this case, the
Hyperthreading property, instead of hyperthreading, may toggle the multi-
core capability on and off.

- IDEController
- IntegratedAudio
- IntegratedSASController*
- IntegratedSATAController*
- IntegratedUSBHub*
- InternalMiniPCI
- KeyboardIllumination*
- LimitCPUIDValue*
- LowPowerS5
- MediaCardAnd1394*

- Microphone*
- ModuleBayDevice*
- MonitorToggling*
- MultiCore

Ø **NOTE:** On some Dell systems, the **MultiCore** property in the **Dell_SMBIOSSettings** class displays **Enabled** or **Disabled** even if the installed processor does not support multi-core technology. In this case, the **MultiCore** property is actually disabled.

- NMIButton*
- NoExecute
- NodeInterleave*
- NumLock*
- Onboard1394
- OnboardModem*
- OSInstallMode*
- ParallelPortConfiguration
- ParallelPortMode
- PasswordBypass*
- PCCard*
- PCCardAnd1394*
- PCISlots
- PenMissingIndication*
- PenResumeOn*
- PointingDevice*
- POSTF12KeySetting
- POSTF2KeySetting
- POST Help Desk Key*
- PowerManagementSettings
- PrimaryParallelATAMaster
- PrimaryParallelATASlave

- PrimaryVideo
- RadioTransmission
- Safe USB*
- SATADIPM*
- SecondaryParallelATAMaster
- SecondaryParallelATASlave
- SerialATAChannel1
- SerialATAChannel2
- SerialATAChannel3
- SerialATAChannel4
- SerialATAChannel5
- SerialATAChannel6
- SerialATAChannel7
- SerialATAChannel8
- SerialCommunication*
- SerialPort1Configuration
- SerialPort2Configuration
- SERRDMIMessage*
- SetCMOSToDefault*
- SingleByteECCLogging*
- SmartCardReader*
- SpeakerVolume
- SpeedStep*
- TabletButtons*
- TertiaryParallelATAMaster
- TertiaryParallelATASlave
- Trusted Platform Module Activation*
- UARTPowerDown*
- USBEmulation

- USBFlashDriveEmulation*
- USBFrontPanelPorts
- USBPorts
- USBPort00*, USBPort01*, USBPort02*, USBPort03*, USBPort04*, USBPort05*, USBPort06*, USBPort07*, USBPort08*, USBPort09*, USBPort10*, USBPort11*, USBPort12*, USBPort13*, USBPort14*, USBPort15*
- USBRearDual*
- USBRearQuad*
- USBWake*
- VideoExpansion*
- VideoMemorySize*
- VT for Direct I/O*
- WakeupOnLAN
- WakeupOnLANMethod
- WiFiCatcherChanges*
- WiFiLocator*
- WirelessDevice
- WirelessLAN
- WirelessSwitchBluetoothControl*
- WirelessSwitchCellularControl
- WirelessSwitchChanges*
- WirelessSwitchWirelessLANControl

NOTE: BIOS settings added in OMCI 7.6 are marked with an asterisk (*).

NOTE: See the *Dell OpenManage Client Instrumentation Reference Guide* on the Dell Support website at **support.dell.com** for detailed information on OMCI classes, properties, and their descriptions.

Glossary

The following list defines or identifies technical terms, abbreviations, and acronyms used in this guide.

ASF
Abbreviation for Alert Standard Format—Defines remote control and alerting interfaces for operating system absent environments.

BIOS
Acronym for basic input/output system. BIOS refers to the software code run by a system when first powered on. The primary function of BIOS is to prepare the system so that other software programs stored on various media (such as hard drives, floppies, and CDs) can load, execute, and assume control of the system. This process is known as booting up.

BIOS can also be said to be a coded program embedded on a chip that recognises and controls various devices that make up the system.

CIM
Acronym for Common Information Model – the CIM Schema is a conceptual schema that defines how the managed elements in an IT environment (for instance Computers or Storage Area Networks) are represented as a common set of objects and relationships between them. CIM is extensible in order to allow product specific extensions to the common definition of these managed elements. CIM uses a model based upon UML to define the CIM Schema.

CIMOM
Acronym for CIM Object Manager. The WMI management infrastructure includes the CIM Object Manager, which is the interface and manipulation point for CIM objects and information. It acts as a facilitator in gathering information and manipulating object properties. The CIMOM stores data in an area called the WMI repository.

CLI

Abbreviation for command line interface. A CLI displays a prompt, the user types a command on the keyboard and terminates the command (usually with the Enter key), and the system executes the command, providing textual output.

CMOS

CMOS which stands for complementary metal-oxide semiconductor, is a major class of integrated circuits. CMOS chips include microprocessor, microcontroller, static RAM, and other digital logic circuits. The central characteristic of the technology is that it only uses significant power when its transistors are switching between on and off states. Consequently, CMOS devices use little power and do not produce as much heat as other forms of logic. CMOS also allows a high density of logic functions on a chip.

COM

Acronym for Component Object Model which is a Microsoft® platform for software componentry. It is used to enable interprocess communication and dynamic object creation in any programming language that supports the technology. The term COM is often used in the software development world as an umbrella term that encompasses the OLE, OLE Automation, ActiveX, COM+ and DCOM technologies. Although it has been implemented on several platforms, it is primarily used with Microsoft Windows®.

DCOM

Acronym for Distributed Component Object Model which is a Microsoft proprietary technology for software components distributed across several networked systems to communicate with each other. It extends Microsoft's COM, and provides the communication substrate under Microsoft's COM+ application server infrastructure.

device driver

A program that allows the operating system or some other program to interface correctly with a peripheral device, such as a printer. Some device drivers—such as network drivers—must be loaded from the **config.sys** file (with a device= statement) or as memory-resident programs (usually from the **autoexec.bat** file). Others—such as video drivers—must load when you start the program for which they were designed.

DLL

Abbreviation for Dynamic Link Library which is Microsoft's implementation of the shared library concept in the Microsoft Windows operating systems. These libraries usually have the file extension DLL, OCX (for libraries containing ActiveX controls), or DRV (for legacy system drivers). The file formats for DLLs are the same as for Windows EXE files—Portable Executable (PE) for 32-bit Windows, and New Executable (NE) for 16-bit Windows. As with EXEs, DLLs can contain code, data, and resources, in any combination.

In the broader sense of the term, any data file with the same file format can be called a resource DLL. Examples of such DLLs include icon libraries, sometimes having the extension ICL, and font files, having the extensions FON and FOT.

DMI

The Desktop Management Interface (DMI) generates a standard framework for managing and tracking components in a desktop, notebook or server computer. Essentially, to the user, it is a table provided by the personal computer BIOS which can be parsed and which gives information about the BIOS and the system in a standardized way. Previously, such complete information was not available from a standardized source in the PC.

Due to the rapid advancement of DMTF technologies, such as Common Information Model (CIM), the DMTF defined an "End of Life" process for its Desktop Management Interface (DMI), which ended 03/31/2005.

DMTF

Abbreviation for Distributed Management Task Force (formerly "Desktop Management Task Force"). DMTF is a standards organisation that develops and maintains standards for systems management of IT environments in enterprises and the Internet. These standards allow building systems management infrastructure components in a platform-independent and technology-neutral way. They provide for systems management interoperability between IT products of different producers, without costly transformations and adaptations.

DOS

Acronym for Disk Operating System.

ECC

Abbreviation for Error-Correcting Code. It is a code in which each data signal conforms to specific rules of construction so that departures from this construction in the received signal can be automatically detected and corrected.

firmware

Software (programs or data) that has been written onto read-only memory (ROM). Firmware can boot and operate a device. Each controller contains firmware that helps provide the controller's functionality.

MAC Address

Acronym for Media Access Control address which is an unique identifier attached to most forms of networking equipment. Most layer 2 network protocols use one of the following three numbering spaces managed by the IEEE: MAC-48, EUI-48, and EUI-64 which are designed to be globally unique.

Magic Packet

In early implementations of Remote Wake-up, the server could be started from a power-off state by sending a Magic Packet. A Magic Packet is an Ethernet packet that contains an adapter's MAC address repeated 16 times in the data field. When an adapter receives a Magic Packet containing its own MAC address, it activates the server's power. This enables network administrators to perform off-hours maintenance at remote locations.

MOF

There are potentially many ways in which CIM management information could be represented to exchange information. The CIM Specification defines a language based on the Interface Definition Language (IDL) called Managed Object Format (MOF).

MOF is the standard language used to define elements of CIM. The MOF language specifies syntax for defining CIM classes and instances. Compilation of MOF files provides developers and administrators with a simple and fast technique for modifying the CIM Repository.

OMCC

Abbreviation for Dell™ OpenManage™ Client Connector. OMCC is a one-to-one management console for local and remote system administration. It is a client management application that allows the user to view and configure OMCI properties and perform remote management functions such as shutdown and BIOS updates.

RAID

Acronym for redundant array of independent disks. In computing, RAID is a system which uses multiple hard drives to share or replicate data among the drives. Depending on the version chosen, the benefit of RAID is one or more of increased data integrity, fault-tolerance and throughput or capacity compared to single drives.

SMART

Acronym for Self-Monitoring, Analysis, and Reporting Technology, which is a monitoring system for system hard disks to detect and report on various indicators of reliability, in the hope of anticipating failures. SMART warns an user or the system administrator of an impending drive failure while time remains to take preventive action, such as copying the data to a replacement device.

SMBIOS

Acronym for System Management BIOS which is a specification to lay out data structures (and access methods) in a BIOS, which allows an user or an application to store and retrieve information specifically about the PC in question.

WBEM

Abbreviation for Web Based Enterprise Management. Defines protocols for the interaction between systems management infrastructure components implementing CIM, a concept of DMTF management profiles that allows defining the behavior of the elements defined in the CIM Schema, the CIM Query Language (CQL) and other specifications needed for the interoperability of CIM infrastructure.

WDM

Abbreviation for the Windows Driver Model in computing. Also known as the Win32 Driver Model, the Windows Driver Model defined a unified driver model for the Windows 98 and Windows 2000 line by standardizing requirements and reducing the amount of code that needed to be written. WDM drivers will not run on operating systems earlier than Windows 98 or Windows 2000 and is designed to be forward-compatible but not backward-compatible.

For example, a WDM driver written for Windows 2000 will load and function under Windows XP, but cannot take advantage of any new WDM features that were introduced in Windows XP. A WDM driver written for Windows XP will not load under Windows 2000.

WfM

Abbreviation for the Intel® Wired for Management standard. WfM technology was used in client-side hardware (including circuitry, memory, power supplies and network interface cards) and management software applications. It defines a baseline set of requirements for managing hardware, including requirements for instrumentation, remote wake-up, power management, and service boot capability. It enables centralized system management, including inventory, fix/repair, configuration, and diagnostics, and provides for off-hours maintenance to minimize downtime. WfM includes support for DMI and allows a technician

to diagnose and upgrade a remote system while the user of that system continues working on other tasks. It also allows software upgrades in the background or during scheduled hours.

WMI

Abbreviation for Windows Management Instrumentation which is a set of extensions to the Windows Driver Model that provides an operating system interface through which instrumented components can provide information and notification. WMI is Microsoft's implementation of the Web-Based Enterprise Management (WBEM) Standard from the Distributed Management Task Force (DMTF).

WMI allows scripting languages like VBScript to manage Microsoft Windows personal computers and servers, both locally and remotely.

Index

A

alerts, 31
  events polled, 35
  local, 32
  remote, 31-32

ASF, 31
  alerting, 32

C

CIM, 7
  cim_iap_data.htm, 42
  cim_win32_data.htm, 42
  overview, 7
  schema, 8

CIMOM, 8, 25

COM, 25

Computer management
  console, 26

D

Data, 8

data provider, 9

data router, 9

DCOM, 12, 45

ddisks.dll, 39

Dell event MOF, 44

Dell instantiation MOF, 44

Dell OMCI MOF, 43

dellWMI.dll, 39

DEvents.dll, 41

DMTF, 31

dpccard.dll, 41

dpci.dll, 41

dsmbios.dll, 40

dsmbios_data.htm, 43

dsysInfo.dll, 41

I

iap.exe, 40

iap_routing_data.htm, 43

indiCons.dll, 42

indiProv.dll, 42

installation
  administrative, 23
  requirements, 17
  silent, 21

instrumentation components, 39